A CELL BASED END USER INTERFACE HAVING ACTION CELLS

RELATED APPLICATION

This is a non-provisional application of provisional application 60/461,772, filed 04/09/03, entitled "Methods and Use of Icons That Represent Elements of a Multi-Window EUI". This non-provisional application claims priority to said '772 provisional application, which specification is hereby incorporated by reference, to the extent it is consistent with the specification and claims to follow.

For the United States of America, this non-provisional application is also a continuation-in-part of co-pending US Application number 10/136,669, filed 04/30/02, entitled "A Cell Based End User Interface", which itself claims priority to:

a) Number 60/287,933, entitled "A MULTI-WINDOW EUI FOR MULTI-MEDIA CONTENT/PROGRAMMING DELIVERY INCLUDING AUTOMATIC RELATIVE RESIZING",

b) Number 60/287,972, entitled "HIERARCHICAL ELEMENTAL STRUCTURE IN SUPPORT OF A MULTI-WINDOW EUI",

c) Number 60/287,663, entitled "BEHAVIOR OF NESTED COMPLEX ELEMENTS",

d) Number 60/287,943, entitled "STATE TRANSITION FOR A MULTI-WINDOW EUI WITH HIERARCHICAL ELEMENTS",

e) Number 60/287,980, entitled "EFFICIENT REGION IMPACT DETERMINATION FOR MULTI-WINDOW EUI",

f) Number 60/287,977, entitled "REGIONS AND ZONE MODIFICATION FOR A MULTI-WINDOW EUI", and

g) Number 60/287,932, entitled "EXPANDABLE CONTROL FACILITY FOR AN END USER INTERFACE";

all filed on 4/30/2001, and which specifications are all hereby fully incorporated by reference, to the extent they are consistent with the specification and claims to follow.

Accordingly, for U.S., the present application, through said '669 non-provisional application, also claims priority to the '933, '972, '663,'943, '980,'977, and '932 provisional applications.

FIELD OF THE INVENTION

The present invention relates to the field of data processing. More specifically, the present invention relates to end user interfaces.

BACKGROUND OF THE INVENTION

With advances in integrated circuit, microprocessor, networking and communication technologies, an end user of a properly equipped television set or a computing device may receive and consume a variety of multi-media contents or programming via a number of different delivery channels. The end user may e.g. receive and consume programming delivered through conventional network broadcast, cable or satellite. The end user may also receive and consume various multi-media contents or programming delivered from various recorded media players, such as VCR tape players, CDROM or DVD players. Alternatively, the end user may also receive and consume various streaming multi-media contents or programming delivered through the Internet or other high-speed digital channel. The end user may also want to work on multiple files, programs, applications or data on their computer.

The end user interfaces (EUI) employed in these multi-media content application interfaces on a computer screen or other display device, Operating System interfaces, or programming deliveries are typically limited in their functionalities and ease-of-use. In particular, they are typically fixed or inflexible, i.e. non-responsive or lack interactivity with the user. For example, in the case of television programming, typically only a single view of a program (chosen by a director) is provided to the end user (even though multiple views are available from the multitude of cameras employed to cover an event or performance). Even at times, when multiple views of a program or windows of an application or multiple different applications have multiple windows open and are provided, the user is unable to change the size, and/or placements of the different display windows within which the views are displayed. Where modifications of the size and/or placement of the display windows are supported (hereinafter, simply windows), typically, automatic relative re-sizing and/or placement of the windows from either unrelated applications, programs or sources across a network or even from a single source, are not supported. That is, expansion of a window will often

result in the blocking of another window (unless the expanding window is a "transparent" window), and contraction of a window will often result in excess unconsumed space (unless the end user takes overt action to enlarge another window). Similar limitations exist in the delivery of multi-media contents, typical computer display interfaces, GUI's of single applications, operating system presentation of multiple windows from multiple applications, sources or programming from recorded media, or windows and/or content associated with streaming through the Internet.

Further, the different windows (whether it is of the same program or of different programs) are usually not easily interchangeable. In particular, associated controls, such as "minimize", "maximize", or task bars, are typically not relocatable from one window associated with one application to another window associated with another application. For example, in the case of television programming, different views of the same program delivered through multiple windows are generally not interchangeable, whereas different programs delivered through different windows, such as a primary view and a "picture-in-picture" (PIP) view, are swappable, provided the end user separately changes the channels associated with the two windows. In the case of windowed applications, control facilities associated with windows of an application, such as "minimize", "maximize" or task bars, typically only alter or affect the corresponding windows of the application, and may not be moved and be associated with another window and/or another application or affect any other application windows or display elements that they do not own (as in the relationship between parent and child windows).

Thus, an improved end user interface for content, individual applications with multiple child windows, desktop display of related or non-related applications, operating system window managers, and/or programming delivery, etc., is desired.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references denote similar elements, and in which:

Figure 1 illustrates an end user view of an EUI implemented in accordance with an embodiment of the present invention;

Figures 2-3 illustrates the anatomy of a cell based hierarchy for implementing the EUI of **Fig. 1**, including the universal region cell, region cells, sub-region cells and zone cells, in accordance with one embodiment;

Figures 4-5 illustrate selected aspects of the composition of a "container" cell, including a region cell and a zone cell, in accordance with one embodiment;

Figure 6 illustrates selected aspects of the composition of an "action" cell, in particular, an icon cell, in accordance with one embodiment;

Figure 7 enumerates selected methods associated with the various implementation cells to support the practice of the present invention, in accordance with one embodiment;

Figure 8 illustrates certain novel end user interface interactions supported under the present invention, by virtual of the architectural design of the hierarchical cell based EUI, in accordance with one embodiment;

Figure 9 illustrates the operational flow of the relevant aspects of an implementor of the present invention, such as an application, a cell manager or a window manager, in support of the novel end user interactions of **Fig. 8**, in accordance with one embodiment;

Figure 10 illustrates the notion of a current view, and the generation of a next view under the present invention, in accordance with one embodiment;

Figures 11-12 illustrate the operational flow of the relevant aspects of an implementor of the present invention, such as an application, a cell manager or a window manager, in support of automatic relative re-sizing or re-placement of region cells or zone cells, in accordance with one embodiment;

Figures 13-14 further illustrate automatic relative re-sizing or re-placement of region cells and zone cells, in accordance with one embodiment;

Figure 15 illustrates the operational flow of the relevant aspects of an implementor of the present invention, such as an application, a cell manager or a window manager, in support of an optimized algorithm for efficiently modifying contiguous region or zone cells, in accordance with one embodiment;

Figure 16 further illustrates the optimized efficient modification of region or zone cells of **Fig. 15**, in accordance with one embodiment;

Figures 17a-17b illustrate two embodiments for practicing the present invention;

Figure 18 illustrate an exemplary computing system or device suitable for practicing the present invention; and

Figure 19 illustrates an exemplary network environment suitable for practicing the present invention.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

Illustrative embodiments of the present invention include, but are not limited to, a hierarchical cell based end user interface, having hierarchically organized display cells (hereinafter, simply cells), processes for the end users to interact with the interface, having particular application to the delivery of multi-media programming and/or content, or having particular application to the organization of multiple windows of different applications running on a single client computer desktop or on multiple display devices for a single or multiple clients, processes for automatically re-sizing and/or repositioning cells of the EUI, components and/or system endowed with some or all aspects of the EUI and the processes.

In the following description, various aspects of the present invention will be described. However, the present invention may be practiced with only some or all aspects of the present invention. For purposes of explanation, specific numbers, materials and configurations are set forth in order to provide a thorough understanding of the present invention. However, the present invention may be practiced without the specific details. In other instances, well-known features are omitted or simplified in order not to obscure the present invention.

Parts of the description will be presented in data processing terms, such as data, variables, methods, requests, returns, and so forth, consistent with the manner commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. As well understood by those skilled in the art, these quantities take the form of electrical, magnetic, or optical signals

capable of being stored, transferred, combined, and otherwise manipulated through mechanical, electrical and/or optical components of a computer system.

The term "display cell" (or "cell" for short) as used herein refers to the logical elements or items employed to collectively implement the various aspects of the EUI. The logical elements/items or cells, as will be described more fully below, are typed and include attributes defining them, including their manifestation and behaviors. Visually, cells may be "nested" within one another. Organizationally, cells may be hierarchically related to each other.

The term "computer system" as used herein includes general purpose as well as special purpose data processing machines, systems, and the like, that are standalone, adjunct or embedded. Examples of general purpose "computer systems" include, but are not limited to, handheld computing devices (palm sized, tablet sized and so forth), laptop computing devices, desktop computing devices, servers, and so forth. Examples of special purpose "computer systems" include, but are not limited to, processor based wireless mobile phones, handheld digital music players, set-top boxes, game boxes/consoles, CD/DVD players, digital cameras, digital CAMCORDERs, and so forth.

Various operations will be described as multiple discrete operations in turn, in a manner that is most helpful in understanding the various embodiments of the present invention, however, the order of description should not be construed as to imply that these operations are necessarily order dependent. In particular, these operations need not be performed in the order of presentation.

The phrase "in one embodiment" is used repeatedly. The phrase generally does not refer to the same embodiment; however, it may. The terms "comprising", "having" and "including" are synonymous, unless the context dictates otherwise.

Figure 1 illustrates an external end user view of an exemplary end user interface **102**, implemented in accordance with one embodiment of the present invention. As illustrated, exemplary end user interface (EUI) **102**, from the end user's perspective, includes multiple display windows **104a-104k**, control facilities **106a-106b** and icons **108a-108j** **(each of which, as will be described in more detail below, may be effectuated as "cells" of the EUI)**. EUI **102** may be

employed to facilitate delivery of multi-media contents or programming for an end user, the presentation of multiple windows from local or remote (across a network or the Internet) applications or the display of data or programs from local or remote applications. An example of such content/programming includes, but are not limited to, presentation of one or more performance or live events, such as sporting events, where the multiple windows are employed to present different views of each performance or event to the end user. An example of the presentation of multiple windows from a single application includes, but are not limited to, an e-mail program that has a window that shows all e-mails available to a user and separate windows for each e-mail that the user has opened and can now respond to, resend or forward to other respondents. An example of multiple applications with simultaneous open windows on one or more desktops or screens (single or multi-screen implementations) includes, but is not limited to, the simultaneous running of one or more spreadsheet programs and one or more word processing programs, each displaying windows on a common display or displays, where data, contents, video, or graphics is being copied and pasted between any two or more separate programs. An example of a local program (client based) and a program that is delivering data from across a network, each displaying windows on a common display or displays includes, but is not limited to, a word processing program resident on a single computer and a browser that is delivering html content from a distant web site. An example of a local program (client based) and a program that is delivering multimedia data from across a network, each displaying windows on a common display or displays includes, but is not limited to, a word processing program resident on a single computer and a videoconferencing system that is delivering video, remote desktops or whitescreen content from other videoconference participants across a network. Networks may include, but not be limited to, a local LAN (Ethernet, ATM, SAN, or other networking technology) or remote (ISND, T1, ATM, Frame, or other networking technology).

For ease of understanding, only a couple of control facilities **106a-106b** and a handful of icons **108a-108j** are illustrated with windows **104a-104k**. As will

be readily apparent to those skilled in the art, based on the descriptions to follow, the present invention may be practiced with more or less of these elements.

More importantly, as will be described in more detail below, EUI **102** is implemented internally via a hierarchy of display cells (or cells for short). The cells are typed and nested. Further, they have attributes, and certain attributes may be inherited in one direction, while others in the other direction, i.e. from a higher level cell, from other cells at the same level, or from a lower level cell (above, at the same level, or below) in the hierarchy. The cells are implemented as data objects with associated methods to facilitate manipulation of their data.

Resultantly, one of the benefits is that the views or windows **104a-104k** are readily controllable by the end user. An end user may select any one of windows **104a-104k**, express a desired modification or change to the size, placement, and/or other related aspects of the windows (such as sound). In response, the implementation logic of embodiments of the present invention, e.g. a cell manager, or alternatively, a window manager or an application itself (not shown), will resize, re-position or otherwise modify the selected windows, as well as all other impacted elements (cells) of EUI **102** accordingly and automatically.

Resizing may be expansion of a selected element or cell of EUI **102**, or contraction of a selected element or cell of EUI **102**. Repositioning of a cell may be within the existing immediately higher-level cell or to another cell of EUI **102**. In various embodiments, control facilities **106a-106b** are provided for the various windows **104*** to facilitate a user in resizing, re-positioning or otherwise modifying the various aspects of the windows **104***.

In one embodiment, as the selected and/or impacted windows **104*** are re-sized, the content of each window **104*** may be automatically scaled, preserving "full" visibility of the contents. That is, the contents of the various windows **104*** remain in full view, scaled, but not truncated or otherwise eclipsed. However, in alternate embodiments, one or more windows **104*** may have their contents truncated or eclipsed, completely encapsulated in another cell or transformed into a completely different cell.

In one embodiment, in addition to being employed for the delivery of multi-media content or programming, one or more of "windows" **104a-104k** may be

employed to present a "pool" of icons (another cell type), each corresponding to an additional displayable or launch-able cell having contents, and/or action that may be performed on the content or the attributes of an associated cell. The former is referred to as an "image icon", and the cell implementing the "image icon" is an image-icon cell, whereas the latter is referred to as a "button icon", and the cell implementing the "button-icon" is a button-icon cell.

These and other aspects of the various embodiments of the present invention will be described more fully below. The asterisk at the end of a reference number denotes a "wild card", representing any of the trailing suffixes of the reference numbers employed in a figure. For example, **104*** stands for **104a, 104b** or any one of the other **104** references of **Fig. 1**.

Figures 2-3 illustrate the relevant aspects of the internal hierarchical cell based implementation of EUI **102** to provide the desired improved features and behaviors, in accordance with one embodiment. As illustrated, in accordance with the present invention, end user interface **102** is cell based, and the constituting cells are nested (**Fig. 2**), and the data objects implementing the cells are hierarchically organized (**Fig. 3**).

As alluded to earlier, cells are typed, and have attributes defining their manifestation and behaviors. Visually, cells may be "nested" within each other. Organizationally, cells may be hierarchically related to each other. The attributes may be inherited in multiple directions, from the higher level cells, cells at the same level of the hierarchy, or from the lower level cells (organizationally speaking).

More specifically, for the embodiment, each EUI **102** is comprised of a number of nested "container" cells and a number "action" cells. For ease of understanding, the "outer most" (from a nesting perspective) or the highest-level (from a hierarchy perspective) "container" cell, that is the cell corresponding to the totality of display space available, within which all other cells are nested, is referred to as the universal or root region cell **202**. Nested within universal region **202** may be one or more nested "container" cells. In particular, at the next highest-level, for the embodiment, for ease of operation, the "container" cells all have visual manifestations that are rectangular in shape, and share borders.

These "container" cells are referred to as regions cells **204a-204c**. In another embodiment, cells of any shape are allowed at any level of the hierarchy.

Selected one or ones of the region "container" cells may further include one or more nested "container" cells. For ease of understanding, these nested "container" cells, except for ones disposed at the "inner most" nesting or "lowest" level (counting only "container" cells), are referred to as sub-region "container" cells **205a-205b**. The "container" cells disposed at the "inner most" nesting or "lowest" level (counting only "container" cells) are referred to as zone "container" cells **206a-206k**. A zone "container" cell **206*** dedicated to the holding of icon "action" cells (to be described more fully later), such as zone "container" cell **206i**, is also referred to as an "icon pool".

"Action" cells, such as those implementing control facilities **208a-208b**, and icons **210a-210j**, whether they are representing other displayable or launch-able cells or merely representing actions to be performed, i.e. image icons or button icons, may be nested within (visually speaking) or descend from (organizationally speaking)) any of the "container" cells, i.e. the universal region cell **202**, such as control facilities cells **208a-208b** and icon cells **210a-210d**, region and sub-region cells **204a-204c** and **205a-205b**, none shown, or zone "container" cells, such as icon cells **210e-210j** or in different structures or elements of the cells themselves (borders, title bars, etc.).

As described earlier, control facilities may include facilities for facilitating minimizing or maximizing an "action" cell, and an icon "action" cell may be an image or a button icon "action" cell. The "container" cell within which another "container" or "action" cell is nested or from which the other "container" or "action" cell is descended, is also referred to as a "host" cell.

Hereinafter, the description will be given with the relationship between the various cells simply be referred to as either being "nested" in another cell or "descended" from another cell, depending on which characterization is more meaningful in view of the context. However, the reference expressed from one perspective (visual or organizational) is an expression in both perspectives, even though expression in the other perspective may not have been explicitly stated.

Continuing now with the description and referring in particular to **Fig. 3**, for the embodiment, the data, such as attribute data (described more fully below), associated with each cell, **202** and **204*-210***, whether "container" or "action", are organized and implemented as an hierarchy of data objects **302** and **304*-306***, with data object **302** corresponding to universal region cell **202** being the root object of the hierarchy, data objects **304*** corresponding to region cells **204*** being descendant data objects of root object **302**, data objects **305*** corresponding to sub-region cells **205*** being descendant data objects of the data objects **304*-305*** of their "host" region/sub-region cells **204*/205***, and data objects **306*** corresponding to zones cells **206*** being descendant data objects of the data objects **302** and **304*-305*** of their host universal/region/sub-region cells **202** and **204-205**.

Contents to be presented in various windows **104***, such as video **308a-308e**, graphics **310a-310b** and texts **312a-312c** are effectuated by associating the data objects of these contents with data objects **306*** of the zone "container" cells **206*** corresponding to windows **104***. Data objects **314a-314h** and **316a-316b** implementing icons **210a-210j** and control facilities **208a-208b** are descendant data objects of the data objects of their respective host universal/regions/sub-regions/zones **202** and **204*-206***.

Resultantly, the novel architecture and data organization enable contents provided through different display windows **104*** to be easily swappable, by swapping the association of the contents' data objects with the "host" zone cell **206***. Similarly, the associations of "action" cells **208*** and **210*** with the different cells **202** and **204*-206*** may also be easily changed, by changing the association between data objects **314*-316*** with data objects **302** and **304*-306*** of cells **202** and **204*-206***.

For ease of understanding, only one zone "container" cell **206a** and limited number of "action" cells **208a** and **210a-210b** are illustrated as being directly nested in universal region **202**, only one region "container" cell **304b** as having sub-region-"container" cells **254***, and only one zone "container" cell **206i** is deployed as an icon pool in **Fig. 2-3**. However, the present invention contemplates multiple nesting of multiple "container" and "action" cells, e.g. more

region/zone "container" cells as well as "action" cells may be nested in universal region **202**, more third level sub-region "container" cells and/or "action" cells may be nested within region "container" cells of the second level. Any nested cell or cells may be represented by an icon or hidden from view (invisible) at any level of the hierarchy. From the description thus far and the ones to follow, those skilled in the art will be able to practice the present invention in such multi-level manner, should that be desired.

Figures 4-5 illustrate the composition of "container" cells, in particular, a region "container" cell and a zone "container" cell, in accordance with one embodiment. From the processing or computation perspective, the earlier described universal region cell **202**, region "container" cell **204***, and sub-region "container" cells **205*** are merely different variants of the region "container" cell to be described. Accordingly, the composition descriptions to follow apply equally to universal region-cell **202**, region "container" cell **204***, and sub-region "container" cells **205***.

As illustrated, for the embodiment, associated with the definition of each region/zone "container" cell **202** and **204*-206***, and stored inside corresponding data objects **302** and **304*-306*** are behavior attributes defining whether a "container" cell **204*-206*** is dynamic or fixed (i.e. created on an as needed basis, or always present), a number of structure attributes defining the structural elements of the "container" cell (kernel **402/502**, boundary **406/506**, etc. (to be described more fully below)), and a number of geometric attributes defining whether the "container" cell's position is movable or stationery, its relative priority to other "container" cells **204*-206***, a center position, a base, a height and a maximum size of the region/zone "container" cells **204*-206***:

region "container" cell	zone "container" cell
region_type = [dynamic, fixed]	zone_type = [dynamic, fixed]
region_position = [stationary, movable]	zone_position = [stationary, movable]
region_priority = [1, 2, 3 ...]	zone_priority = [1, 2, 3 ...]
region_center_position	zone_center_position
region_base	zone_base

region_height	zone_height
region_maximum_size	zone_maximum_size

Additionally, for the embodiment, associated with the definition of each region/zone "container" cell **202** and **204*-206***, and stored inside corresponding data objects **302** and **304*-306*** are additional geometric attributes further defining kernel **402/502** of the region/zone "container" cell **204*-206***. A kernel of a region/zone "container" cell **204*-206*** refers to the smallest manifestation of the region/zone "container" cell **204*-206***. That is, when the available space within a host "container" cell **202-205*** falls below the space required by the kernel of a region/zone "container" cell **204*-206***, the "container" cell **204*-206*** is to be "reduced" to an icon cell. For the embodiment, the kernel related geometric attributes include attributes defining a region/zone "container" cell's kernel's size, base and height.

region-cell	zone-cell
region_kernel_area	zone_kernel_area
region_kernel_base	zone_kernel_base
region_kernel_height	zone_kernel_height

Further, for the embodiment, associated with the definition of each region/zone "container" cell **202** and **204*-206***, and stored inside corresponding data objects **302** and **304*-306*** are geometric attributes further defining a boundary **406/506** of the region/zone "container" cell **204*-206***. The boundary may also have related attributes including geometric/visual attributes defining a thickness and a color of the boundary of the region/zone "container" cell **204*-206***.

region-cell	zone-cell
region_boundary_thickness	zone_boundary_thickness
region_boundary_color	zone_boundary_color

In one embodiment, if the "boundary" attributes are not specified for a region/zone "container" cell, the region/zone "container" cell automatically inherits the "boundary" attributes of the nearest "ancestor" region "container" cells, where such attributes are specified. In other words, an inheriting region/zone "container" cell takes on the characteristics of the bequeathing "ancestor" region "container" cell.

Associated with the definition of each region/zone "container" cell **202** and **204*-206***, and stored inside corresponding data objects **302** and **304*-306*** may also include geometric attributes further defining a border **404/504** of the region/zone "container" cell **204*-206***. The border related attributes include geometric/visual attributes defining a thickness, a color, a texture, a shading, a blinking and a transparency attribute of the border of the region/zone "container" cell **204*-206***.

region-cell	zone-cell
region_border_thickness	zone_border_thickness
region_border_color	zone_border_color
region_border_texture	zone_border_texture
region_border_shading	zone_border_shading
region_border_blinking	zone_border_blinking
region_border_transparent	zone_border_transparent

In one embodiment, if the "border" attributes are not specified for a region/zone "container" cell, the region/zone "container" cell also automatically inherits the "border" attributes of the nearest "ancestor" region "container" cell, where such attributes are specified.

In various embodiments, for a region "container" cell **204*-205***, the attributes may further include policy attributes defining how many zone "container" cells it may have, their names and their default alignments (e.g. center, top, bottom, right, left and so forth), whereas for a zone "container" cell **206***, the attributes may further include a relationship attribute defining its "host" region "container" cell **202** and **204*/205***. For a zone "container" cell **206***, the

attributes may further include content attributes defining its content types, video, data, image, text, and so forth, and an external buffer **508**.

In various embodiments, the policy attributes may also associate an internal buffer(s) 409, 509 with variable algorithms that may be initiated to change a cell's attributes, priorities, policies or behaviors when the cell itself decreases in size and the cell border touches or encounters the internal buffer edge. The policy attributes may also associate internal buffer(s) 409, 509 with variable algorithms that may be initiated to change other cells attributes, priorities, policies, or behaviors at any level of the hierarchy when the cell itself decreases in size and the cell border touches or encounters the internal buffer edge. Further, the policy attributes may also associate an internal buffer(s) 409, 509with variable algorithms that may be initiated to change any section of the hierarchy, the entire hierarchy, the policies and behaviors of a section of or the entire hierarchy, or elements, applications, data external to the hierarchy or system itself, when the cell itself decreases in size and the cell border touches or encounters the internal buffer edge.

In various embodiments, the overlapping of two structures (collections of nested cells), e.g. the overlapping of one or more cells of the structures, may be referred to as a "violation" (if the policy attributes of the structural elements so define). The policy attributes may associate an internal buffer with a variable algorithm that may be initiated to cause the entire hierarchy to collapse and be reduced to a single icon, in the event of a "violation". In different embodiments, the "violation" of structural elements, such as an internal buffer of a single cell "violating" an external-buffer or other structural element of another cell, may cause a wide variety of responses or changes to the hierarchy of cells or external to the hierarchy including causing cells to iconize, shrink, swap, freeze in size or position, grow, become visible, transparent, or invisible, or move to other locations, cells to change relative priorities, pop-up cells to become visible, policy or structural changes anywhere in the system or hierarchy, the communication of an alarm, multimedia channel(s) to open, an application or applications to start on a client or remote server, a local or remote script to run that alters any aspect of the client, network, remote device(s), a mechanical switch to be thrown, or any

other action allowable by the system to a portion of or the complete hierarchy of cells (in accordance with policy or policies governing the processing of the "violations". External buffer 508, on the other hand, may act like internal buffers or may define the minimum inter-zone "container" cell spacing between immediately adjacent zone "container" cells 206*. Similarly, the policy attributes of external buffer 508 may be associate the buffer with a variable algorithm that may be initiated to changes the cell's attributes, priorities, behaviors based on the cells that are within the external buffer area or touch the external buffer edge. The policy attributes may also associate external buffer 508 with a variable algorithm that may define an "area of influence" that alters the attributes, priorities, behaviors of other cells at any level of the hierarchy within the external buffer area or touch the external buffer edge based on algorithms or policies.

region-cell	zone-cell
region_zone_list = [zone-cell names]	zone_region_association
region_zone_alignment = [center, top, bottom, right, left]	zone_video, zone_data, zone_image, zone_text
region_max_allowable_zones = [number]	

The above described attributes for region/zone "container" cells are merely illustrative. In alternate embodiments, the present invention may be practiced with more or less region/zone "container" cell attributes. For example, the present invention may be practiced with additional attributes defining

a) the control facilities associated with the region/zone "container" attributes,

b) the behavior when certain areas of a region/zone "container" cell is "mouse over", and

c) forced bequeathing of certain attributes to the more inner or lower level region/zone "container" cells.

Further, the attributes of a nested container cell may be inherited from any of its more outer container cells.

Figure 6 illustrates the composition of an "action" cell, more specifically, an image icon "action" cell in further detail, in accordance with one embodiment. As described earlier, an image icon "action" cell is an iconic representation of another displayable or launch-able cell with content, control facilities and so forth. Further, "action" cells may also include cells defining control facilities, and cells defining "button" icon, which provide control facilities for a region/zone "container" cell and action to be performed within a region/zone "container" cell respectively. The description to follow for an image icon "action" cell may be likewise adopted to implement button icon "action" cells and/or control facilities cells.

As illustrated, for the embodiment, associated with the definition of each image icon "action" cell **208*-210*** and stored inside a corresponding data object **314*-316*** are geometric, visual and relationship attributes defining e.g. the bit map of the image icon "action" cell, and the center position of the image icon "action" cell, a region/zone "container" cell with which the image icon "action" cell is associated, and a buffer **602**. Buffer **602** defines the minimum space required to display the image icon "action" cell.

Icon-cell
image_icon_association = [region/zone-cell name]
image_icon_center_position = [x, y]
image_icon_actual = [bit_map_name]
image_icon_buffer_base = [ ]
image_icon_buffer_height = [ ]
image_icon_upper_left_vertex_position = [x, y]

Similarly, in alternate embodiments, the present invention may be practiced with more or less attributes (structural, geometric, visual, policy, behavior, and so forth) defining the various "action" cells, as well as the contents to be rendered (i.e. video, graphics, texts, and so forth). The attributes may also be inherited from the immediate as well as higher hierarchical level host container cells.

In particular, for button icon "action" cells and control facility "action" cells, each of the respective "action" cells may include one or more behavior attributes in identifying the binaries to be executed responsive to various types of user actions, e.g. "mouse over", "single click", "double clicks", and so forth.

The binaries may specify one or more actions to be performed. The exact actions to be performed are application dependent. However, the one or more actions may e.g., cause one or more contents to be rendered, one or more contents to be processed, one or more container cells to instantiated and be added among the currently active and visible container cells, one or more scripts to be executed, one or more defining attributes to be added to, modified, or deleted from the defining attributes of one or more cells, one or more cells to be added to or deleted from the cell hierarchy of the EUI, one or more user inputs to be requested, or one or more system actions be taken.

Additionally, the one or more actions may be performed in association with or on behalf of one or more container cells. The one or more container cells may be any container cells in any level of the cell hierarchy of the EUI. In other words, in embodiments of the present invention, the one or more actions to be performed need not be confined to the immediate host container cell of the action cell (icon).

Further, the one or more actions to be performed may also be conditioned on one or more policy attributes of a host container cell, and/or one or more state values of a container cell, which may or may not be the same container cell on which's policy attributes the action(s) to be performed is(are) conditioned. That is, the binaries may be designed in a manner that the various possible actions are conditionally performed based on one or more attribute and/or state values of the one or more container cells, with or on behalf of which, the actions are to be performed.

In other words, under embodiments of the invention, an action cell (icon) may be employed e.g., to represent one or more regions, sub-regions and/or zones, and have the one or more regions, sub-regions, and/or zones to be instantiated and added among the currently visible regions, sub-regions and/or zones. An action cell (icon) may be employed to cause contents to be processed

in association with or on behalf of these regions, sub-regions or zones, or rendered in the regions, sub-regions or zones. An action cell (icon) may be employed to cause inputs to be requested of a user, or actions requested of a system, in association with or on behalf of these regions, sub-regions or zones.

An action cell (icon) may also be employed to cause regions, sub-regions, zone or icons be added or deleted from the EUI. An action cell (icon) may also be employed to modify a region, sub-region, or a zone, including adding, modifying or deleting defining attributes of the regions/sub-regions/zones. More importantly, these actions may be taken in association with, or on behalf of any regions, sub-regions or zones disposed at any level of the cell hierarchy of the EUI.

Referring briefly to **Fig. 3** again, as described earlier, for the illustrated embodiment, region/zone "container" cells, "action" cells, and data (include video, graphics, text and so forth) are implemented in an object oriented manner, with corresponding data objects **302** and **304*-316***. In one embodiment, as illustrated in **Figure 7**, various methods **700** are associated with the data objects **302** and **304*-316***. For the embodiment, these methods include in particular a clear, a contract, an expand, a remove, and a set attribute method, **702-710**, associated with the root data object **302**, and inherited by the descendant data objects **304*-316*** of the nested region/zone "container" cells **204*-206***, as well as the descendant data objects **314*-318*** of the nested "action" cells **208*-210***.

Clear method **702**, when invoked against universal region "container" cell's data object **302** clears the EUI **102**, i.e. removing all nested region/zone "container" cells **204*-206***, including their contents, as well as any nested "action" cells **208*-210***. In one embodiment, the universal region "container" cell clearing is efficiently achieved by clearing or deleting all descendant data objects **304*-316***. Inner invocation against a region/zone "container" cell **204*-206*** clears the nested regions/zones "container" cells **204*/206*** within the target region/zone "container" cell **204*** including their contents, and any nested "action" cells **208*-210***. In like manner, the clearing is efficiently achieved by clearing or deleting the applicable descendant data objects **304*-316***.

Expand and contract methods **704-706** are employed to expand and contract a region/zone "container" cell **204*-206*** respectively. Remove method **708** facilitates removal of individual cells of the EUI **102**, i.e. one or more regions/zone "container" cells **204*-206*** or "action" cells **208*-210*** without clearing all cells. Removal is achieved in like manner as clear method **702**, except the operation is applied to selected ones of the descendant data objects, as opposed to all descendant data objects. Set Attribute method **710** facilitates setting of the earlier described region/zone "container" cell and "action" cell attributes associated with region/zone "container" cells **202** and **204*-206***, and "action" cells respectively.

For region/zone cells **204*-206*** and "action" cells **208*-210***, their corresponding data objects **304*-306*** and **314*-316*** further include the association of a create, and a delete, a move and a place method **712-718**. Create and delete methods **712-714**, as their names suggest, facilitate creation and delete of the various descendant data objects **304*-306*** and **314*-316*** for the nested region/zone "container" cells and "action" cells **204*-206*** and **208*-210***. Move and place methods **716-718**, as their names suggest, facilitate movement and relocation of the various region/zone "container" cells and "action" cells **204*-206*** by modifying e.g. the position attributes of the corresponding data objects **304*-306*** and **314*-316***.

For the embodiment, data objects **314*-316*** for "action" cells **208*-210*** further include the association of a launch method **720** for launching a displayable region/zone cell **204*-206*** represented by image icon "action" cells **210***.

With the exception of the handling of the impact that flows from the creation, deletion, expansion and contraction of a region/zone "container" cell **204*-206***, implementation of the above described methods are within the ability of those ordinarily skilled in the art, accordingly will not be further described. Handling of the impact that flows from the creation, deletion, expansion and contraction of a region/zone "container" cell **204*-206*** will be described in more detail below, referencing **Figures 11-16**.

Figures 8-9 illustrate two novel interactions with EUI **102**, otherwise not available under the prior art, and the relevant operation flow of an implementor, such as an application, a cell manager or a window manager, incorporated with the teachings of the present invention. As illustrated, by virtue of the earlier described novel architecture and data organization, contents presented through two different zone "container" cells **206*** may be easily interchanged or swapped, as denoted by arrow **802**. The swapping operation may be initiated through any one of a number of user key sequences, e.g. user key sequences similar to a conventional drag and drop operation. The swapping may be efficiently accomplished by switching association of the applicable data objects **308*-316*** and their region/zone "container" cells **204*-206***. Further, "action" cells **314*-316*** may be easily relocated to any region/zone "container" cell **204*-206*** as denoted by arrow **804**.

As illustrated in **Fig. 9**, in response to a non-region/zone "container" cell impacted user interaction, an implementor (such as an application, a cell manager or a window manager incorporated with the teachings of the present invention) determines if the sequence of user inputs denotes a drag and drop of content from one region/zone "container" cell **204*-206*** to another, block **902**. If so, the implementor effectuates the content swapping, by switching the data objects' association with their region/zone "container" cells **204*-206***, as earlier described, block **904**.

If the sequence of user inputs does not denote a drag and drop of content, for the embodiment, the implementor further determines if the sequence of user inputs denotes an "action" cell drag and drop, block **906**. If so, the implementor effectuates the "action" cell movement and placement by similarly switching the "action" cell's association with region/zone "container" cells **204*-206***, optionally launching the represented region/zone "container" cell **204*-206*** and its contents (if so requested by the sequence of user inputs), block **908**.

If the sequence of user inputs does not denote either one of these novel interactions supported, the denoted prior art request may then be processed as in the prior art.

The sequence of user inputs denoting the earlier described content and "action" cell drag and drop may be practiced through any key sequences, e.g. by clicking on the content or icon, using a cursor control device, and keeping the applicable click button of the cursor control device held down, until the target region/zone "container" cell **206*** is reached. At such time, the click button of the cursor control device may be returned to its normal position. In alternate embodiments, the present invention may be practiced with other key sequences instead.

Figure 10 illustrates an overview of the operation of EUI **102**. The current state of EUI **102** as defined by the current states of the corresponding data objects **302-316*** of the constituting cells **202-210*** of EUI **102**, as illustrated, is referred to as the current view of EUI **102**. In response to user interactions, such as a request to add or remove a region/zone "container" cell **204*-206***, or a request to expand or contract a region/zone "container" cell **204*-206***, the implementor of the present invention, e.g. an application, a cell manager or a window manager, performs a series of responsive calculations, and generate the next view of EUI **102**.

The operational flow of the relevant aspects of the implementor, in response to the various user interactions of interest, will be described in turn below.

Figure 11 illustrates the operational flow of the relevant aspects of an implementor, e.g. an application, a cell manager or a window manager, for responding to a request to add a region/zone "container" cell or an "action" cell to a region/zone "container" cell, or expand a region/zone "container" cell (hereinafter, for the description of **Fig. 11**, simply the "add/expand" request), in accordance with one embodiment. As illustrated, for the embodiment, the implementor first determines if the requested addition or expansion fits in the current available space of the host region/zone "container" cell, block **1102**. The required space to accommodate the requested addition/expansion may e.g. be determined from the attribute values of the "new" or expanded region/zone "container" cell. If the requested addition or expansion fits in the current available

space of the host region "container" cell, the requested addition or expansion is performed accordingly, block **1104**.

However, if the requested addition or expansion does not fit in the current available space of the host region/zone "container" cell, the implementor successively undertakes one or more space creation actions, until either sufficient amount of available space has been created or until all possible space creation actions have been exhausted, blocks **1102-1108**. As soon as sufficient available space has been created, operation continues at block **1104** as earlier described.

However, if all possible space creation actions have been exhausted and the amount of space required to accommodate the requested addition or expansion remains insufficient, the implementor successively undertakes one or more space requirement reduction actions, until either the required space has been reduced below the amount of available space or until all possible space reduction actions have been exhausted, blocks **1110-1112**. Similarly, as soon as the required space to satisfy the addition or expansion request is reduced below the available space, operation continues at block **1104** as earlier described.

If likewise, all possible required space reduction actions are exhausted, and the amount of space required to accommodate the add/expand request remains above the available space, an "error", such as "unable to add/expand", is returned in response to the request.

In one embodiment, available space creation actions include shifting existing region/zone "container" cells within the host region/zone "container" cell the add/expand request is to be performed, and reducing the existing region/zone "container" cells if necessary. In one embodiment, shifting of existing region/zone "container" cells includes shifting the existing regions/zone "container" cells to a predetermined corner of the host region/zone "container" cell, e.g. the lower left corner, the upper left corner, the upper right corner or the lower right corner. In one embodiment, shifting of existing region/zone "container" cell to a corner is performed by aligning the region/zone "container" cells along one or the other boundary forming the corner. In another embodiment, shifting of existing

region/zone "container" cells to a corner is performed by alternating in aligning the regions/zone "container" cells along the boundaries forming the corner.

In one embodiment, reducing the existing region/zone "container" cells is performed in an incremental manner. In another embodiment, reducing the existing region/zone "container" cells is performed in accordance with the relative priorities of the existing region/zone "container" cells. In one embodiment, reduction is performed in an incremental manner as well as in view of the relative priorities of the existing region/zone "container" cells. In one embodiment, the lowest priority region/zone "container" cell is first successively reduced to its kernel before the next higher priority region/zone "container" cell is successively reduced towards its kernel. In another embodiment, the reduction is successively performed in a round robin manner. In yet another embodiment, reduction of existing region or zone "container" cells further includes reducing one or more of the existing region/zone "container" cells to their icon "action" cell representations. Again, in one embodiment, the reduction to iconic representation is performed in view of the relative priorities of the existing region/zone "container" cells.

In one embodiment, reduction of required space action includes successively reducing the size of the region/zone "container" cell to be added, or to be expanded to.

Still referring to **Figure 11**, back at block **1104**, upon performing the requested addition/expansion, the implementor determines if any post addition/expansion operations need to be performed. If so, the post addition/expansion operations are performed, block **1118**. If not, the process terminates.

Post addition/expansion operations may be required, as existing region/zone "container" cells may have been shifted to one corner of the host region/zone "container" cell or reduced, even to their kernel, in the course of accommodating the addition/expansion request. Accordingly, for the embodiment, upon accommodating the addition/expansion, attempts are made to at least partially restore the shifted and/or reduced region/zone "container" cells back to the pre-request state. Similarly, the post addition/expansion operations

may include successively expanding reduced existing region/zone "container" cells, which may also be performed in view of the relative priorities, re-shifting shifted region/zone "container" cells (e.g. out from the coalesce corner) to achieve a more balance alignment of the nested region/zone "container" cells within the host region "container" cell. "Balance" may be measured e.g. by the average space gap between the boundaries of the various region/zone "container" cells.

Figure 13 illustrates an exemplary addition of a new region/zone "container" cell into a region "container" cell having two existing region/zone "container" cells, in accordance with the above described process. As illustrated, the two existing region/zone "container" cells are first shifted to the lower left corner, with the two existing region/zone "container" cells aligned along the left boundary forming the lower left corner (illustrations A & B). Since there isn't enough available space to add the requested new region/zone "container" cell, the two existing region/zone "container" cells are successively reduced, eventually to their kernels, first the lower priority region/zone "container" cell, then the higher priority region/zone "container" cell (illustrations C-D). The new region/zone "container" cell is then added to the newly created space in the opposite upper right corner (Illustration E). Further, the reduced region/zone "container" cells are shifted back out from the lower right corner and aligned in the bottom portion of the host region/zone "container" cell (illustration F & G).

Figure 14 illustrates another exemplary addition of a new region/zone "container" cell into a region/zone "container" cell having three existing region/zone "container" cells, in accordance with the above described process. Except in this illustration, the existing region/zone "container" cells are first shifted to the upper left corner, and then shifted out along the top portion of the host region/zone "container" cell instead. Further, the new region/zone "container" cell is reduced to reduce its space requirement before it is added to the host region/zone "container" cell.

Figure 12 illustrates the operational flow of the relevant aspects of an implementor, e.g. an application, a cell-cell manager or a window manager, for responding to a request to remove a region/zone "container" cell from a region

"container" cell, or an "action" cell from a region/zone "container" cell (hereinafter, for the description of **Fig. 12**, simply the "remove/contract" request), in accordance with one embodiment. As illustrated, for the embodiment, the implementor removes or contracts the region/zone "container" cell, or the "action" cell as requested, block **1202**. Thereafter, the implementor determines if the there are iconized region/zone "container" cells of the host region/zone "container" cell that can be restored into the newly increased available space of the host region/zone "container" cell, block **1204**. If so, the implementor restores one or more of the eligible iconized region/zone "container" cells, subject to the available space, block **1206**. In one embodiment, the restoration is performed in accordance with the relative priorities of the iconized region/zone "container" cells.

Upon exhausting the possibility of restoring iconized region/zone "container" cells (either because there are none left or there isn't enough space), the implementor determines if there are any reduced region/zone "container" cells that can be grown towards their maximum sizes, block **1208**. If so, the implementor successively grows one or more of the reduced region/zone "container" cells, subject to the available space, block **1210**. In one embodiment, the successive growth is also performed in accordance with the relative priorities of the reduced region/zone "container" cells.

Next, similar to the process of adding or expanding a region/zone "container" cell, upon restoring or growing the iconized or reduced region/zone "container" cells, the implementor determines if any post restoration or growth actions need to be performed, block **1212**. If so, the implementor performs the post restoration or growth actions, such as shifting and aligning to "re-balance" the region/zone "container" cells of the host region/zone "container" cell, block **1214**. As before, "balance" may be measured e.g. by the average space gap between the boundaries of the various region/zone "container" cells

Figure 15 illustrates the operational flow of the relevant aspects of an implementor, e.g. an application, a cell manager or a window manager for responding to a request to expand a region "container" cell (hereinafter, for the description of **Fig. 15**, simply the "add/expand" request), in accordance with

another embodiment. In this embodiment, for efficiency of operation, region "container" cells are nested within a host region "container" cell in a contiguous manner, i.e. without available space gap between their boundaries.

As illustrated, in response to a request to grow a region "container" cell by an amount, the implementor first generates extended boundaries for the growth region "container" cell (see **Fig. 16**), block **1502**. Next, the implementor determines growth impact for up to n levels removed in all directions, using the extended boundaries.

For example, for the exemplary growth request illustrated in **Fig. 16**, growth impact of the center region "container" cell may be determined using its extended boundaries, based on their intersections with other boundaries. The impacts on region "container" cells up to 2 degrees removed from the center region "container" cell may be summarized as follows:

	up	down	left	right
Neighbor region "container" cell affected	A, B	F, E	H, G	C, D
Second level region "container" cell affected	none	none	special case	L, K, J
Side Effects	H, C	D	F	none

Thereafter, for the embodiment, the implementor iteratively expands the region "container" cell in the various directions, adjusting the impacted region "container" cell to accommodate the growth, block **1506**. The process continues until the desired amount of growth is achieved. If the desired growth is not achievable, for the embodiment, an "error", such as "growth unachievable", is returned, block **1508**.

As alluded to earlier, the present invention may be practiced e.g. by endowing an application itself, a cell manager or a window manager with the teachings of the present invention. In the latter cases, a cell/window manager

implementor may be effectuated in at least two manners, **Fig. 17a** and **Fig. 17b**. In the embodiment of **Fig. 17a**, cell manager **1704** is equipped with teachings of the present invention interfaces and interacts with applications **1702** using its services, and display device driver **1706** as in the prior art. Accordingly, under this embodiment, typically, the universal region "container" cell **202** is the entire display space of a display device.

In the alternate embodiment of **Fig. 17b**, the cell manager implementor operates as an "auxiliary" cell manager **1703** to a conventional window manager **1704**. Applications **1702** may interact with conventional window manager **1704** directly or indirectly through auxiliary cell manager **1703** (equipped with the teachings of the present invention). Accordingly, universal region "container" cell **202** may be a window of a conventional window approach, except within that window, the EUI is implemented and practiced as earlier described, in accordance with the present invention.

In yet other alternate embodiments, auxiliary cell manager **1703** may be integrally incorporated as part of window manager **1704**.

Figure 18 illustrates an exemplary computer system or device suitable for practicing the present invention, in accordance with one embodiment. As shown, computer system/device **1800** (hereinafter simply "device") includes one or more processors **1802** and system memory **1804**. Additionally, device **1800** includes mass storage devices **1806** (such as diskette, hard drive, CDROM and so forth), input/output devices **1808** (such as keyboard, cursor control and so forth) and communication interfaces **1810** (such as network interface cards, modems and so forth). The elements are coupled to each other via system bus **1812**, which represents one or more buses. In the case of multiple buses, they are bridged by one or more bus bridges (not shown). Each of these elements performs its conventional functions known in the art. In particular, system memory **1804** and mass storage **1806** are employed to store a working copy and a permanent copy of the programming instructions implementing the implementor of the present invention, e.g. an application, a cell manager or a window manager. The permanent copy of the programming instructions may be loaded into mass storage **1806** in the factory, or in the field, through a distribution medium (not

shown) or through communication interface **1810** (from a distribution server (not shown)). The constitution of these elements **1802-1812** are known, and accordingly will not be further described.

Figure 19 shows an exemplary network environment suitable for practicing the present invention, in accordance with one embodiment. In this embodiment, contents are presented for user of client device **1902** to enjoy, employing the hierarchical cell based EUI **102** of the present invention. In one embodiment, display device **1904a** on which EUI **102** is rendered, is an integral of client device **1902**. In another embodiment, display device **1904b** on which EUI **102** is rendered, is an separate and distinct "peripheral" of client device **1902**.

In various embodiments, the implementor of the present invention, e.g. an application, a cell manager or a window manager, may be executing on client device **1902** itself. In other embodiments, the implementor may be executing on server **1906** instead. Examples of the former case may be a personal computer, an enhanced integrated television set, and a set-top box. Examples of the latter case may be a content streaming server or a cable programming broadcasting device.

Client device **1902** and server **1906** are coupled to each other via one or more private and/or public networks, including e.g. the Internet, employing Digital Subscriber Lines (DSL) (or other variants xDSL), Cable Network, Integrated Digital Service Network (ISDN), Asynchronous Transfer Mode (ATM), Frame Relay, or other high performance communication links/connections of like kind. Communications between client device **1902** and server **1906** may be accomplished via any one of a number of communication protocols known in the art, including but are not limited to the TCP/IP protocol.

Examples of content may include one or more of the following content or program types

Special Events	News	TV	Sports
Concerts	Live	Reality	Golf
	Produced		
Olympics	Shows	Network	Football
Political			
Rallies		Syndication	Racing
Amusement		Children's TV	Football
Plays		Treasure Hunts	Soccer

Thus, a novel EUI method and apparatus has been described. Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that a wide variety of alternate and/or equivalent implementations may be substituted for the specific embodiments shown and described, without departing from the scope of the present invention. This application is intended to cover any adaptations or variations of the embodiments discussed herein. Therefore, it is manifestly intended that this invention be limited only by the claims and the equivalents thereof.